Machine Learning in Updating Predictive Models of Planning and Scheduling Transportation Projects

LIYE ZHANG AND W. M. KIM RODDIS

A method combining machine learning and regression analysis to automatically and intelligently update predictive models used in the Kansas Department of Transportation's (KDOT's) internal management system is presented. The predictive models used by KDOT consist of planning factors (mathematical functions) and base quantities (constants). The duration of a functional unit (defined as a subactivity) is determined by the product of a planning factor and its base quantity. The availability of a large data base on projects executed over the past decade provided the opportunity to develop an automated process updating predictive models based on extracting information from historical data through machine learning. To perform the entire task of updating the predictive models, the learning process consists of three stages. The first stage derives the numerical relationship between the duration of a functional unit and the project attributes recorded in the data base. The second stage finds the functional units with similar behavior—that is, identifies functional units that can be described by the same shared planning factor scaled in terms of their own base quantities. The third stage generates new planning factors and base quantities. A system called PFactor built on the basis of the three-stage learning process shows good performance in updating KDOT's predictive models.

Planning and scheduling transportation projects is an interesting and important subarea of transportation engineering. Construction project planning and scheduling can be viewed in two stages. The first stage involves planning and scheduling of project development and engineering, a task that is typically the responsibility of transportation agencies. This stage deals with planning and scheduling project preparation going on in the agency itself before release of the project for bid. It stresses the allocation of resources within the agency. The second stage involves planning and scheduling of project execution and construction, a task that is typically the responsibility of the contractors. This paper deals with the first stage of planning and scheduling transportation projects.

The accuracy of predicting duration required by activities of a project will influence to a great extent the effectiveness of planning and scheduling the project because failure to manage time properly will result in schedule slippage and cost overruns. But most predictive models of activity duration for planning and scheduling of transportation projects by state departments of transportation are based on experience in a relatively ad hoc manner and often do not accurately reflect the agency's current business practices and requirements. Establishing new predictive models for activity duration in order to improve planning and scheduling is of concern to state departments of transportation.

Predictive models are mathematical relations. Manually updating predictive models is theoretically possible but practically infeasible because of the complexity of this real-world engineering problem. It not only uses both numeric and symbolic data types, but also is multidimensional and nonhomogeneous. This paper presents a method combining machine learning and regression analysis to automatically and intelligently update predictive models used in determining the durations of transportation projects.

First, this paper briefly outlines the method of planning and scheduling transportation projects. Second, it discusses the predictive models to be updated. Then, following a brief review of the machine learning methods, the method combining machine learning and regression analysis is presented. Finally, the performance of the PFactor system, built using this method, is given.

PLANNING AND SCHEDULING

The Kansas Department of Transportation (KDOT) manages many types of transportation projects. Generic planning templates are available for typical project types such as bridge replacement, new road construction, and pavement overlay. To plan and schedule a transportation project in its management network, KDOT follows the steps shown in Figure 1.

For a new transportation project, the project statement is given to a planner. First, the planner analyzes the transportation project according to its project statement, identifying all activities that must be performed in order to complete the project. The planner chooses a generic template that most closely matches the project type, establishing an activity network for the project. Next, the duration of each activity in the project is estimated according to the predictive models stored in the management system. Finally, a complete plan and schedule are generated either by forward or backward pass calculation (*1*).

For a typical template, an activity network flow chart is used and the critical path method (*1*) is adopted. Figure 2 shows the activity network flow chart of a generic template suitable for a simple bridge replacement project.

The basic parts of the management network for a project are *work phases*, *events*, and *activities*. Work phases are made up of events and activities. Events are either milestones or border check points (less-significant milestones). The components of the management network are shown in Figure 2. For instance, the utility work phase comprises the events of UTILP (utility plans), UTAGR (utility agreement complete), and UTCOM (utility adjustment complete), and the activities of UTENG (utility engineering) and UTADJ (utility adjustments). The milestones and border check points are the

Department of Civil Engineering, University of Kansas, Lawrence, Kan. 66044.

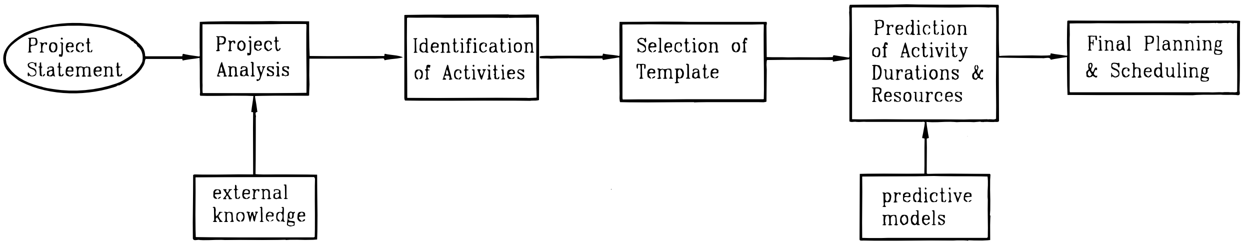

Project Statement
Project Analysis
external knowledge
Identification of Activities
Selection of Template
Prediction of Activity Durations & Resources
predictive models
Final Planning & Scheduling

FIGURE 1 Process of managing a project.

beginning or ending of an activity and mark a particular point in time for reference or measurement. They do not take any elapsed time in planning and scheduling.

Activities are associated with time, and time is their important factor. An activity can start when all predecessors to that activity are complete. For example, the activity of PS&E (plans, specification, and estimates) in Figure 2 can start only when its predecessors of UTADJ (utility adjustments), FIDES (final design), and RWCDM (right-of-way condemnation) are finished.

An activity consists of subactivities called *functional units*. As indicated in Table 1, the activity of DSSUR (design surveys) includes the functional units of FSURV (field survey), DESUP (design support), DMATL (district materials), and MATLS (materials). The functional units of an activity can be performed at the same time, and their durations may be different. Therefore, the duration of an activity is determined by the functional unit whose duration is the longest. The total duration of a project is determined by the summation of the time taken by the activities on critical path (*1*). The critical path is defined as the longest continuous chain of activities through the network schedule that establishes the minimum overall project duration.

It is clear that the more accurate the duration prediction of functional units, the more accurate the duration prediction of activities. Consequently, improved duration prediction results in more effective planning and scheduling.

PREDICTIVE MODELS

The duration of functional units of a project strongly depends on the attributes of the project. (The approach described does not explicitly consider the interplay between cost and duration, reflecting KDOT practice. Further work addresses this issue but is beyond the scope of this paper.) The attributes of a project include road length, number of lanes and bridges, and location of a project. The attributes used in describing a project in KDOT's data base system are given in Table 2, which indicates that a project has many attributes and that the attributes are of mixed types—that is, both symbolic and numeric. In terms of project attributes, the duration of a functional unit can be described as

$$d = f\{\text{attributes}\} \tag{1}$$

where d denotes the duration of the functional unit, and f is a mathematical function of attributes. However, instead of all attributes in Table 2, in general, only several of those attributes influence a particular functional unit. To establish the predictive model for the duration of a particular functional unit, the experts in planning and scheduling, drawing on their experiences, determine the following:

- What attributes influence the duration of the particular functional unit? That is, what attributes are the significant independent attributes on which the duration of the functional unit depends? Different functional units may have different significant attributes.
- How do the attributes determine the duration of the particular functional unit? That is, what is the numerical relationship between the significant attributes and the dependent duration of the functional unit?

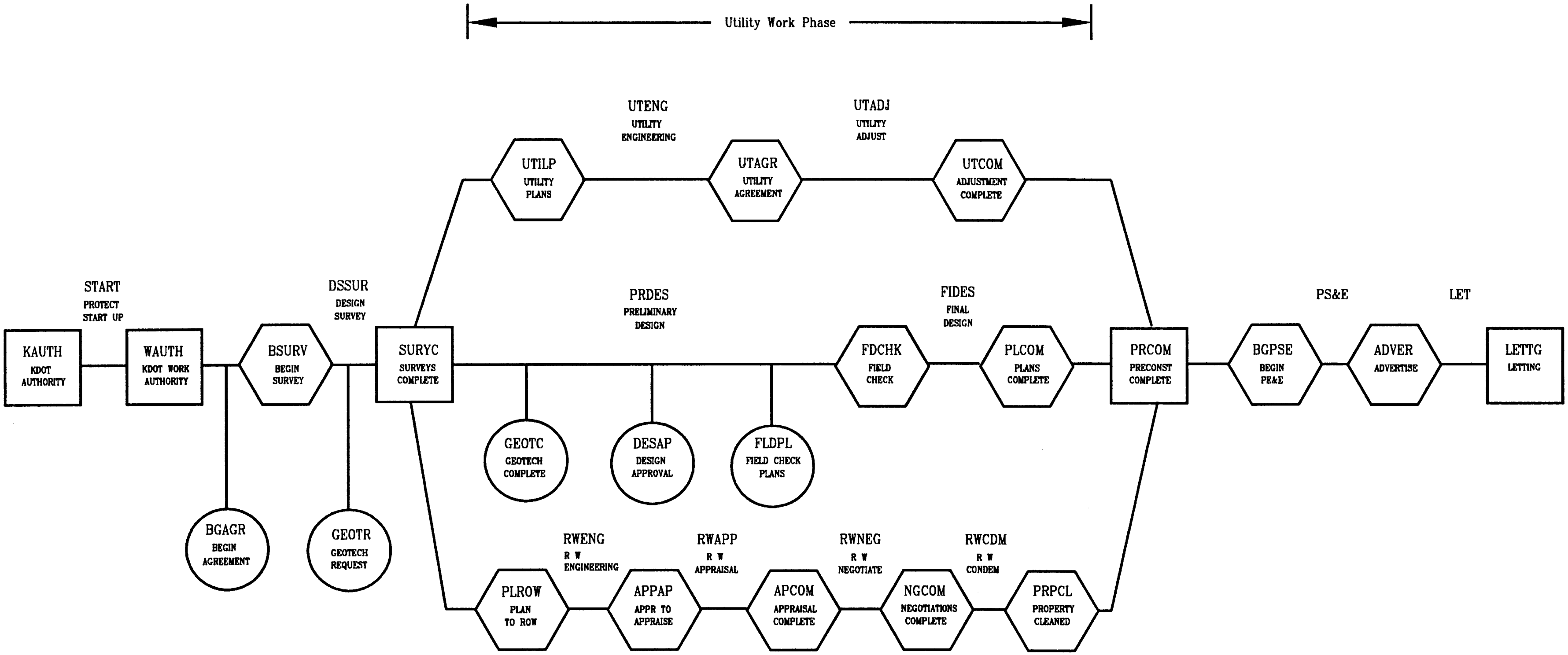
KDOT manages many transportation projects of a variety of types. Many templates are stored in the management system to classify the various projects. There are hundreds of project type associated functional units in the management system. Individually predicting the duration of each functional unit would lead to an excessive number of models. However, engineers in KDOT observed that some functional units behave similarly, with durations differing only by a constant; that is, those functional units have the same significant attributes and those significant attributes influence the duration in the same way except for magnitude. This can be accounted for in the predictive model by splitting the duration of a particular functional unit into two parts, B and p, with the duration of the functional unit measured by the product of B and p.

$$d = B \times p \tag{2}$$

where B is a constant related to the functional unit and independent of the attributes, and p is a function of the significant attributes. B and p are called a *base quantity* and a *planning factor*, respectively, in KDOT's planning and scheduling management system. In other words, predictive models consist of planning factors and base quantities. The duration of functional units is proportional to their corresponding planning factors with the base quantities as the constants of proportionality.

The introduction of planning factors in KDOT's planning and scheduling system allows the system to predict the duration of many functional units in terms of a small number of planning factors. Some functional units even in different templates may share the same planning factor by having their own base quantities.

The preceding description applies to the predictive models in use at KDOT. However, these models have become outdated and no longer accurately reflect the agency's current business practices and requirements. There is thus a need to update the predictive models. A large data base of planning and scheduling information is available on projects executed over the past decade. It would be useful for KDOT to have an automated method of updating the predictive models on the basis of historical data. Such an automated approach could then be used continuously to incorporate information contained in recent additions to the data base.



The order of events significance:

1. square - major importance
2. hexagon - moderate importance
3. circle - least importance

FIGURE 2 Activity network flow chart of generic template.

TABLE 1 Functional Units Contained in Activities in Template of 3R/Bridge Replacement

Activities	Functional Units
START	START
DSSUR (DESIGN SURVEY)	FSURV (FIELD SURVEY) DESUP (DESIGN SUPPORT) DMATL (DISTRICT MATERIALS) MATLS (MATERIALS)
PREDES (PRELIMINARY DESIGN)	BRIDG (BRIDGES) ENVIR (ENVIRONMENTAL) GEOL (GEOLOGY) PVMNT (PAVEMENT) SOILS (SOILS) ROAD (ROAD DESIGN)
FIDES (FINAL DESIGN)	BRIDG (BRIDGES) TRCO (TRAFFIC CONTROL) ENVIR (ENVIRONMENTAL) GEOL (GEOLOGY) LANDS (LANDSCAPE) PVMT (PAVEMENT) ROAD (ROAD DESIGN) SIGNS (SIGNING) SOILS (SOILS)
RWENG (RIGHT OF WAY ENGINEERING)	RWENG (RIGHT OF WAY ENGINEERING)
RWAPP (RIGHT OF WAY APPRAISAL)	RWAPP (RIGHT OF WAY APPRAISAL) CONOF (CONSTRUCTION OFFICE)
RWNEG (RIGHT OF WAY NEGOTIATIONS)	CONOF (CONSTRUCTION OFFICE) RWACQ (RIGHT OF WAY ACQUISITIONS) RQMGT (RIGHT OF WAY MANAGEMENT) RWREL (RIGHT OF WAY RELOCATIONS)
RWCDM (RIGHT OF WAY CONDEMNATION)	LEGAL (LEGAL)
UTENG (UTILITY ENGINEERING)	UTIL (UTILITIES) CONOF (CONSTRUCTION OFFICE)
UTADJ (UTILITY ADJUSTMENT)	CONOF (CONSTRUCTION OFFICE)

THREE-STAGE PROCESS OF UPDATING PREDICTIVE MODELS

To allow the updated predictive models to be integrated easily with KDOT's existing planning and scheduling system, it is preferred that the format of predictive models remain unchanged, which means keeping predictive models in the same form of planning factors and base quantities.

The analysis described in this paper is limited to linear estimates, reflecting KDOT practice. This limitation prevents the use of exponential forms used in many areas of construction to reflect scale economies (or diseconomies). Current research is under way to allow other than linear forms, but that work is beyond the scope of this paper.

Each model is constructed to predict the duration of a functional unit. Therefore, the updating process begins at the level of functional units. The updating process consists of the following three stages:

- Analyzing the data set of each functional unit,
- Grouping functional units with similar behavior, and
- Generating new planning factors and base quantities.

The most difficult part of updating predictive models is the first stage, that is, finding the numerical relations between the functional unit duration and its significant attributes. The difficulties come from the following characteristics of the problem:

1. Attributes are of mixed types, symbolic and numeric.
2. The numerical relations between duration and attributes are multidimensional and nonhomogeneous: different relationships hold in different subsets of the data set of the functional unit, which are expressed as region-equation pairs

$$R_i : d = f_i\{\text{attributes}\} \tag{3}$$

where

i = region number,
R_i = description of region i, and
f_i = numerical relation of region i.

To ensure that the updated models are in a form that is clear to the users, the data analysis is guided by knowledge specific to the planning and scheduling problem area. This domain knowledge is

TABLE 2 Attributes Related to Planning Factors

ATTRIBUTE NAME	ATTRIBUTE VALUES	TYPE
Access indicator	Controlled or Uncontrolled	Symbolic
Borrow	Yes or No	Symbolic
Bridges	The number of bridges	Numeric
Bridge replacement	The number of bridge replacements	Numeric
Bridge width	Numeric	Numeric
Bridge length	Numeric	Numeric
Construction under traffic	Yes or No	Symbolic
Crossing	Small, Medium, Large	Symbolic
Design	In-House or Consultant	Symbolic
Distance	Travel miles from Topeka	Numeric
FHWA improvement type	Integer indicating different types	Symbolic
Lanes	Two, Four or Six	Symbolic
Length	Numeric	Numeric
Light tower	The number of light tower required	Numeric
Location study	Major, No-major	Symbolic
Location construct	New or Existing	Symbolic
Metro	Normal or High	Symbolic
Places	Kansas City, Wichita, Topeka or Others	Symbolic
Relocation	Yes or No	Symbolic
Sign footing	The number of sign footings required	Numeric
Sign project	New or Modified	Symbolic
Sign truss	Yes or No	Symbolic
Surface work type	Grading & surfacing , Grading or Surfacing	Symbolic
Surface material	Bituminous or Concrete	Symbolic
Time	Time of letting: Jan., Feb.,..., Dec.	Symbolic
Tracts	The number of tracts to be purchased	Numeric
Tracts relocated	The number of relocated tracts in negotiation	Numeric
Tracts condemned	The number of relocated tracts in condemnation	Numeric
Urban indicator	Urban or Rural	Symbolic
US81 indicator	East or West	Symbolic
US283 indicator	East or West	Symbolic
Utilities	The number of utilities	Numeric
Utilities relocation required	Yes or No	Symbolic

derived largely from the existing models, which encode the expertise of planners and schedulers. The important restrictions used in guiding the data analysis include (*a*) the function describing each region is limited to linear functions as discussed at the beginning of this section, and (*b*) region boundaries are solely dependent on symbolic attributes. Therefore, the expected forms of numerical relationships can be expressed by

$$R_i : d = c_{i_0} + c_{i_1}a_1 + c_{i_2}a_2 + \ldots + c_{i_m}a_m \qquad (4)$$

where $a_1, a_2, \ldots, a_m$ are numerical attributes; and $c_{i_0}, c_{i_1}, c_{i_2}, \ldots, c_{i_m}$ are region-related constants. Different regions may have different significant attributes.

3. Significant attributes are unknown before data analysis. That is, it is unknown before data analysis what symbolic attributes should be used in region descriptions and what numerical attributes should be used in numerical equations. When many attributes are present, choosing the most significant attributes is a computationally intensive task, even if a linear function of the significant attributes in each region is required.

These problem characteristics preclude a straightforward application of traditional statistical regression analysis. Traditional statistical regression analysis must assume a model a priori (unlike Characteristic 3 just given). It requires variables of one type (unlike Characteristic 1), and also requires that the numerical relations be homogeneous, that is, the same relationship is true over the entire domain (unlike Characteristic 2).

In addition, the second and third stages of updating predictive models are time-consuming and computationally expensive because of the large number of functional units. Manually updating predictive models is theoretically possible but practically infeasible. For these reasons, the approach combining machine learning and regression analysis is applied to update predictive models intelligently and automatically.

COMBINATION OF MACHINE LEARNING AND REGRESSION ANALYSIS

Machine learning is the subfield of artificial intelligence concerned with the design of automatic procedures able to learn from training cases. Since the early 1950s when Turing (*2*) proposed this application for computers, machine learning from examples has been an area of research (*3*). Since the 1980s, machine learning has made substantial progress, and various machine learning methods have been proposed. They can be classified into five paradigms.

The first paradigm uses decision rules, decision trees, or similar knowledge representations. One of the successful algorithms in this paradigm is a tree-based method named C4.5 developed by Quinlan (*4*). A limitation of these methods is their requirement of discrete values for attributes. When the predicted decision is in the form of ordered continuous numeric values instead of finite classes, the proposed algorithms include CART (constructing regression trees) and M5 (generating model trees) (*5,6*).

The second paradigm is case-based or instance-based learning. Rather than extracting from the examples some abstract such as trees and storing this structure in memory, these methods store instances or cases in memory, and classify unseen cases by referring to similar remembered cases. The group contains methods such as nearest-neighbor algorithms (*7*), *k*-nearest-neighbor algorithms (*8*), and average-case analysis (*9*).

The third paradigm is neural networks. They represent knowledge as a multilayer network of threshold units that spreads activation from input nodes through internal units to output nodes. Therefore, the knowledge, such as mathematical functions, hidden in the data is not explicitly represented. A comprehensive presentation of various neural networks is given by Langley and Iba (*10*).

The fourth paradigm is genetic algorithms, which was derived from the evolutionary model of learning (*11*). Genetic algorithms use the Darwinian principle of "survival of the fittest." A genetic classifier is composed of a set of classification elements that replicate and mutate to form new generations. The more successful elements produce variants of themselves and proliferate, whereas elements performing poorly are discarded. BEAGLEs outlined elsewhere (*11*) are the example systems in this group.

These four paradigms typically attempt to improve the accuracy of classification and prediction. The fifth paradigm concerns numeric law discovery. Systems such as ABACUS (*12*) and IDS (*13*) were developed from the BACON algorithm (*14*), which was designed to discover scientific laws on the basis of empirical data evidence. BACON systems attempt to find an invariant based on the variables given as input in order to build the model iteratively. But the BACONs appear better able to explain historical laws with artificial data than to discover new ones. A critical review of these methods can be found elsewhere (*15*). Another system, KEPLER, was suggested by Wu and Wang (*16*). These systems are domain-independent but have requirements for data bases such as small size, free of noise, and one function covering whole domain space.

The particular task at hand of updating predictive models stresses not only the improvement of prediction accuracy for new cases, but also the explicit representation of knowledge hidden in data as mathematical functions. To reach these two goals, a three-stage learning algorithm is used.

The first-stage learning process is to find the relationship between duration and attributes. Because of the difficulties mentioned before, the algorithm combines machine learning techniques and statistical analysis to complete the learning efficiently.

This algorithm uses a tree-based model (called M-model tree) as its knowledge representation of region-equation pairs. This knowledge representation fits the application domain and is able to describe clearly hidden relationships as shown in Figure 3. Region descriptions are expressed by the nodes and arcs of the tree. Regional numerical relationships are expressed by the linear equations in the tree leaves.

The algorithm starts by dividing the data set into training and testing sets. The training set *T* is used for building an M-model tree. The testing set is used for assessing the M-model tree and controlling pruning.

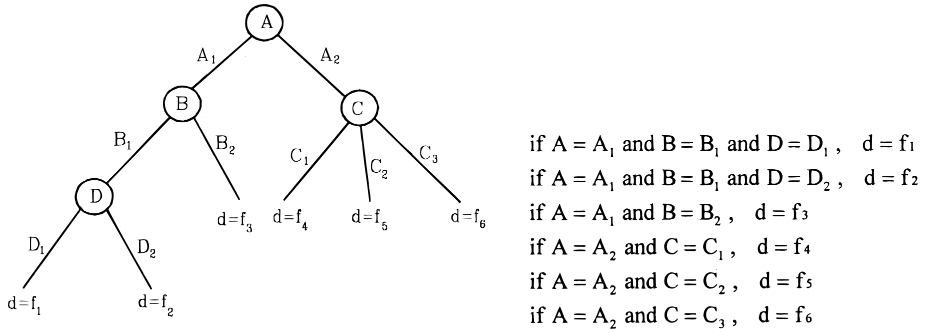
The first step of building an M-model tree is to compute the standard-deviation (*17*) of the target values of the cases in *T* that is treated as a measure of error. Unless *T* contains very few cases or its measure of error is less than a threshold, *T* is split into two or more subsets T_i on the basis of one of the symbolic attributes in order to make the training cases in the subsets more homogeneous. The criterion to select an attribute as a node of the M-model tree is evaluated by the expected error reduction (*5,6*)

$$\Delta\, error = SD(T) - \sum\nolimits_{T}^{T_i} SD(T_i) \tag{5}$$

where $SD\ (T)$ denotes the standard deviation of the set of training case *T*, and $SD\ (T_i)$ denotes the standard deviation of the subset of training cases T_i. The algorithm uses a greedy search to choose the symbolic attribute that maximizes the expected error reduction. This process is repeated on the subsets until either every subset contains few cases or the error measure is less than a threshold. Only symbolic attributes not used before can be selected for the current node.

Multivariate linear models are constructed for the cases at each node of the M-model tree, using standard regression analysis (*17*). However, instead of using all numeric attributes in the standard regression analysis of each node, the numeric attributes used in the equation of a node are restricted to the numeric attributes inherited from its parent node.

After each linear model is obtained, it is simplified by eliminating numeric attributes to minimize its weighted standard deviation.



if A = A_1 and B = B_1 and D = D_1 , d = f_1
if A = A_1 and B = B_1 and D = D_2 , d = f_2
if A = A_1 and B = B_2 , d = f_3
if A = A_2 and C = C_1 , d = f_4
if A = A_2 and C = C_2 , d = f_5
if A = A_2 and C = C_3 , d = f_6

FIGURE 3 Tree representation of region-equation pairs.

Weighted standard deviation of a node is defined as $\Sigma\,(T_i/T)\,SD\,(T_i)$ after a symbolic attribute is selected for the node. This algorithm uses a greedy search to remove attributes whose elimination decreases the weighted standard deviation. In some cases, the algorithm may remove all numeric variables, leaving only a constant at the leaf.

In the process of building the M-model tree, only training cases are used. Testing cases are used to prune the M-model tree in order to simplify the tree to give better prediction without overfitting. Each nonleaf node of the model tree is examined, starting just above the leaves after the M-model tree is built up. The algorithm chooses as the final model for this node either the simplified linear model or the model subtree, depending on which has the lower error estimate on the testing data. If the linear model is chosen, the subtree at this node is pruned to a leaf.

The algorithm just described is applied to all subdata bases of functional units. The completed first stage generates a forest consisting of M-model trees as shown in Figure 4.

In the second stage of learning, the M-model trees are compared to figure out which trees are similar so that their corresponding functional unit can be described by the same planning factors. Two trees are similar if

- The tree structures are the same: if, tree leaves are in one-to-one correspondence, the attributes used in corresponding nodes are the same, and the attribute values in the corresponding arcs are the same.
- The numeric equations in corresponding leaves are proportional, that is (see Figure 3)

$$\frac{Eq\_21}{Eq\_11} = e_1 \qquad \frac{Eq\_22}{Eq\_12} = e_2 \qquad \frac{Eq\_23}{Eq\_34} = e_3 \tag{6}$$

- The ratios of the proportionality of equations are constant, that is,

$$e_1 = e_2 = e_3 = \text{constant} \tag{7}$$

The third stage of learning divides the trees in groups on the basis of their similarities. For each group, one tree is selected as the primary tree. The ratio B of a tree to the primary tree is the base quantity for the functional unit corresponding to the tree. The planning factor for the group is the average of the trees divided by their corresponding base quantities.

SYSTEM PERFORMANCE

Using the algorithm discussed previously, a system called PFactor has been implemented. The performance of the system is discussed here using two example cases, one consisting of an artificial data set and one consisting of an actual engineering data set.

The first case simulates the real project data base with the data built from known functions with a known noise level. The case is used to show PFactor's whole learning process deriving planning factors and base quantities from data. This data set is divided into three subdata sets, each of which consists of 200 examples. Every example has 10 independent variables $x_1, \ldots, x_{10}$ and one dependent variable y. The data were generated from the following models by Matlab:

Take $x_1, \ldots, x_5$ symbolic independent attributes. The discrete values of these attributes are distributed evenly, that is,

$$\begin{aligned}
&P(x_1 = Y) = P(x_1 = N) = 1/2\\
&P(x_2 = T) = P(x_2 = F) = 1/2\\
&P(x_3 = E) = P(x_3 = W) = 1/2\\
&P(x_4 = R) = P(x_4 = S) = P(x_4 = T) = 1/3\\
&P(x_5 = A) = P(x_5 = B) = P(x_5 = C) = 1/3
\end{aligned} \tag{8}$$

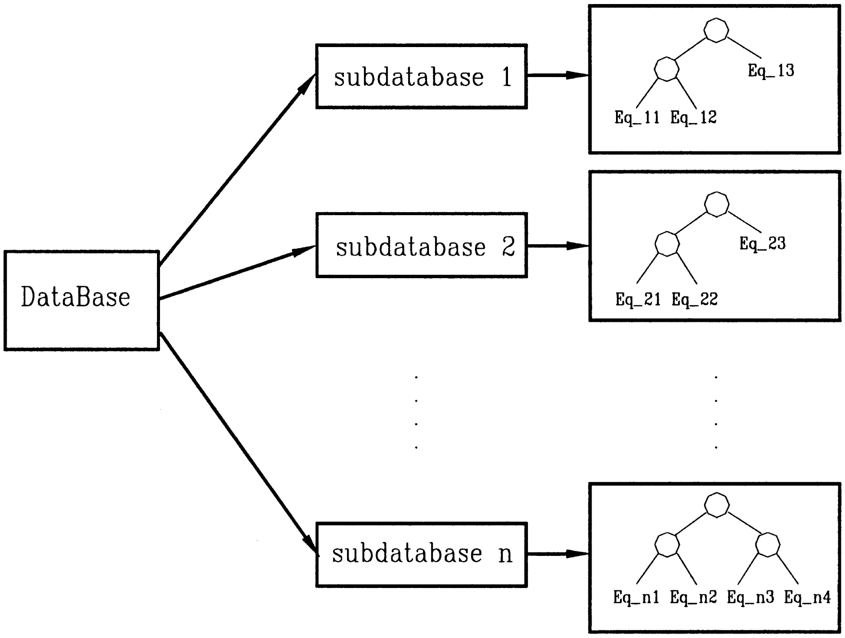


FIGURE 4 Results of first stage of learning process.

Take $x_6, \ldots, x_{10}$ numeric independent attributes that have values in the range of 0 to 1. Let Z, introduced noise, be independent of $x_1, x_2, \ldots, x_{10}$ and normally distributed with mean 0 and variance 2. Then the first subdata set implies

$$\begin{array}{ll} \textit{if } x_1 = Y \text{ and } x_2 = T, & \text{set } y = 5 + 8x_6 + 20x_7 + Z \\ \textit{if } x_1 = Y \text{ and } x_2 = F, & \text{set } y = 6 + 4x_6 + 10x_7 + Z \\ \textit{if } x_1 = N, & \text{set } y = 5 + 8x_8 + Z \end{array} \quad (9)$$

The second subdata set implies

$$\begin{array}{ll} \textit{if } x_1 = Y \text{ and } x_2 = T, & \text{set } y = 10 + 16x_6 + 40x_7 \\ & \quad + Z = 2(5 + 8x_6 + 20x_7) + Z \\ \textit{if } x_1 = Y \text{ and } x_2 = F, & \text{set } y = 12 + 8x_6 + 20x_7 \\ & \quad + Z = 2(6 + 4x_6 + 10x_7) + Z \\ \textit{if } x_1 = N, & \text{set } y = 10 + 16x_8 \\ & \quad + Z = 2(5 + 8x_8) + Z \end{array} \quad (10)$$

The third subdata set implies

$$\begin{array}{ll} \textit{if } x_1 = Y \text{ and } x_2 = T, & \text{set } y = 8 + 6x_6 + 14x_7 + Z \\ \textit{if } x_1 = Y \text{ and } x_2 = F, & \text{set } y = 8 + 10x_6 + 15x_8 + Z \\ \textit{if } x_1 = N, & \text{set } y = 12 + Z \end{array} \quad (11)$$

The first two subdata sets behave similarly. They can be described by the same region-equation pairs. If the equations of the first subdata set are selected, the second subdata set should be described by the first model multiplied by the constant 2. The output results show that there are two planning factors. The first planning factor is

$$\begin{array}{ll} \textit{if } x_1 = Y \text{ and } x_2 = T, & pf = 4.51 + 8.06x_6 + 19.82x_7 \\ \textit{if } x_1 = Y \text{ and } x_2 = F, & pf = 5.76 + 4.03x_6 + 9.55x_7 \\ \textit{if } x_1 = N, & pf = 5.37 + 7.38x_8 \end{array} \quad (12)$$

The second planning factor is

$$\begin{array}{ll} \textit{if } x_1 = Y \text{ and } x_2 = T, & pf = 8.00 + 6.20x_6 + 13.67x_7 \\ \textit{if } x_1 = Y \text{ and } x_2 = F, & pf = 8.33 + 9.17x_6 + 15.25x_8 \\ \textit{if } x_1 = N, & pf = 12.00 \end{array} \quad (13)$$

where *pf* stands for *planning factor*. The first subset can be described by planning factor 1 and base quantity 1; the second subset can be described by planning factor 1 and base quantity 2.10; the third subset can be described by planning factor 2 and base quantity 1. Those are the results one would expect.

The second case is a real engineering data set. This case is used to show that the updated predictive models generated by PFactor provide better duration prediction than the existing models. The performance of duration prediction is measured by the percentage deviation, which is defined as the average over the testing cases of the ratio of the residual to the target duration value.

To simplify this case, the data set was selected to modify only planning factor No. 18. The data set consists of 179 examples of functional unit RWAPP (right-of-way appraisal) from three project templates. The existing models for duration prediction use the planning factor No. 18 and base quantities (equal to 10) for this functional unit in the templates. Therefore, only one model tree and one base quantity should be generated. When the data set is generated from the master project data base, domain engineers use the domain knowledge to exclude the attributes irrelevant to duration of the functional unit. In this case, nine independent attributes are left. Five of the attributes are symbolic, and four are numeric:

Symbolic: ⟨US_81⟩, ⟨Lanes⟩, ⟨Urban_ind⟩, and ⟨Util_reloc⟩
Numeric: ⟨Length⟩, ⟨Bridges⟩, ⟨Tracts⟩, ⟨Tracts_condem⟩, and ⟨Tracts_reloc⟩

PFactor generates the tree as shown in Figure 5. The tree shows that only two symbolic attributes and two numeric attributes significantly affect the duration. The significant symbolic attributes are selected by the error reduction, and the insignificant numeric attributes are eliminated by the weighted standard deviation.

The tree is generated as follows. At the root node, PFactor calculates the error reduction of all symbolic attributes. It finds that the attribute ⟨Util_reloc⟩ gives the maximum error reduction, therefore the attribute ⟨Util_reloc⟩ is used in the root node. Next, the algorithm calculates the weighted standard deviation and finds that no numerical attributes reduce the weighted standard, so no numerical attributes are eliminated. All numerical attributes will be used in its child nodes. For the No branch of ⟨Util_reloc⟩, PFactor again calculates the error reduction of all remaining symbolic attributes and selects ⟨US81_ind⟩. At the node ⟨US81_ind⟩, the algorithm calculates the weighted standard deviation and finds that the attributes ⟨Bridges⟩, ⟨Tracts_condem⟩, and ⟨Tracts_reloc⟩ reduce the weighted standard deviation, so those attributes are eliminated and only the attributes ⟨Length⟩ and ⟨Tracts⟩ are used in its child nodes. At the leaf level, no weighted standard deviation can be obtained, so numerical attributes are further eliminated only when their elimination does not significantly influence the standard deviation. For example, on the branch ⟨US81_ind⟩=East, the algorithm finds that eliminating the attribute ⟨Tracts⟩ influences the standard deviation within a preset threshold of 15 percent, therefore the attribute ⟨Tracts⟩ is eliminated and only the attribute ⟨Length⟩ remains. This model tree does not continue growing beyond this point because the remaining symbolic attributes ⟨Lanes⟩ and ⟨Util_reloc⟩ do not have enough examples in one of their branches to allow further splitting. The other branches are grown down to leaves in a similar manner. PFactor also tries to prune the tree, but in this case the results show that it is unnecessary. According to the tree in Figure 5, the derived new planning factor is

$$\begin{array}{l} \text{if} \langle \text{Util\_reloc} \rangle = \text{No and } \langle \text{US81\_ind} \rangle = \text{East}, \\ \hfill pf = 83.89 + 7.19 * \langle \text{Length} \rangle \\ \text{if} \langle \text{Util\_reloc} \rangle = \text{No and } \langle \text{US81\_ind} \rangle = \text{West}, \\ \hfill pf = 66.75 + 8.75 * \langle \text{Length} \rangle \\ \text{if} \langle \text{Util\_reloc} \rangle = \text{Yes}, \\ \hfill pf = 77.33 + 4.64 * \langle \text{Tracts\_condemned} \rangle \end{array}$$

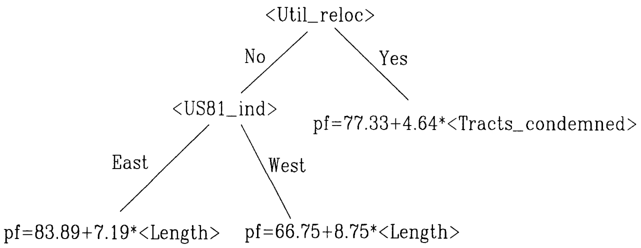


FIGURE 5 Tree structure of planning factor in second example.

The base quantity is 1. Comparing the old and new predictive models, the percentage deviation of the new predictive models is 50.7 percent, while the percentage deviation is 67.5 percent using the old planning factor. This example shows wide deviation for both models, but the new model does improve the percentage deviation of the duration prediction by 25.4 percent based on the current data quality.

The percentage deviation 50.7 percent of the new predictive models is still high because of the noise introduced by the method of drawing the data set from the master data base. An example of such noise is the fact that the actual duration is obtained from the data base by the following calculation:

$$\textit{duration} = \text{end\_date} - \text{start\_date} \tag{14}$$

If a functional unit was suspended for some time, the calculation does not exclude the time when no work was done on the functional unit. This results in the duration of the functional unit used to update the models being greater than the actual duration. On the other hand, the more people working on a project, the shorter the duration. When different numbers of people work on a functional unit, the functional unit will have different durations based on the method of drawing data Equation 14. However, duration obtained from the data base does not include the information on how many people work on functional units of a project. In other words, inconsistent duration units degrade the quality of the data set. Better predictive models are expected to be obtained if the quality of data sets is improved.

The further work is focusing on two aspects of obtaining better predictive models: cleaning the data sets to reduce noise and improve quality, and relaxing the restriction to linear models.

CONCLUSION

The PFactor system built up on the algorithm discussed in this paper is able to update automatically the predictive models of planning factors and base quantities used by KDOT using historical data. The predictive models updated by PFactor improve the prediction accuracy over the existing models. This improvement was achieved even though serious difficulties were encountered concerning data quality. The developed technique for updating predictive models based on information extracted from machine learning can be applied broadly to improve prediction.

ACKNOWLEDGMENTS

This research was funded in part by the Kansas Transportation Research and New Developments Program. Computation resources were provided through the DesignLab under a National Science Foundation grant.

REFERENCES

1. Stella, P. J., and T. E. Glavinich. *Construction Planning and Scheduling*. Associated General Contractors of America, 1994.
2. Turing, A. Computing Machinery and Intelligence. *Mind*, Vol. 59, 1950.
3. Cohen, P. R., and E. A. Feigenbaum. *The Handbook of Artificial Intelligence*, Vol. 3. William Kaufmann, Inc., 1982.
4. Quinlan, R. J. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publisher, Inc., 1994.
5. Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regress Tree*. Wadsworth, Belmont, Calif., 1984.
6. Quinlan, R. J. Learning with Continuous Classes. *Proc., AI '92* (Adams and Sterling, eds.), 1992.
7. Dasarathy, B. V., ed. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, Calif., 1991.
8. Stanfill, C., and D. Waltz. Toward Memory-Based Reasoning. *Communications of the ACM*, Vol. 29, 1986, pp. 1213–1228.
9. Skapura, D. M. *Building Neural Networks*. ACM Press, 1996.
10. Langley, P., and W. Iba. Average-Case Analysis of a Nearest Neighbor Algorithm. *Proc., 13th International Joint Conference on Artificial Intelligence*, Vol. 2, 1993.
11. Forsooth, R. *Machine Learning, Principles and Techniques*. Chapman and Hall Company, 1989, Ch. 4.
12. Falkenhainer, B. C., and R. S. Michalski. Integrating Quantitative and Qualitative Discovery: The Abacus System. *Machine Learning*, Vol. 1, No. 4, 1986.
13. Nordhausen, B., and P. Langley. A Robust Approach to Numeric Discovery. *Proc., 7th International Conference on Machine Learning* (B. W. Porter and R. J. Mooney, eds.), Morgan Kaufmann Publisher, Inc., 1990.
14. Langley, P., H. A. Simon, G. L. Bradshaw, and J. M. Zytlow. *Scientific Discovery, Computational Explorations of the Creative Processes*. MIT Press, Cambridge, Mass., 1987.
15. Schaffer, C. *Domain-Independent Scientific Function Finding*. Ph.D. dissertation. Rutgers University, New Brunswick, N.J., 1990.
16. Wu, Y. H., and S. Wang. Discovering Functional Relationships from Observations Data. In *Knowledge Discovery in Databases* (G. Piatetsky-Shapiro and W.J. Frawley, eds.), MIT Press, Cambridge, Mass., 1991.
17. Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, England, 1992.

Publication of this paper sponsored by Committee on Artificial Intelligence.